ENGINEERING 12: PHYSICAL SYSTEMS ANALYSIS
LABORATORY 3
NUMERICAL ANALYSIS

OBJECTIVES

1. To model the coupled pendula system from the previous lab using two numerical integration methods: Euler’s method and Runge-Kutta fourth-order.

2. To characterize the accuracy and stability of both methods, and to compare the resulting data to the analytic model from the previous lab.

THEORY

In this lab, we will learn two numerical methods for approximating the solutions to ordinary differential equations. Before we begin, we review the Taylor series, an important mathematical concept which will allow us to construct and analyze these methods.

Taylor series. Any mathematical function \( f(x) : \mathbb{R} \to \mathbb{R} \) can be expressed as a polynomial series whose coefficients depend only upon the function value and its derivatives at a particular point \( x_0 \):

\[
f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{6}f'''(x_0) + \ldots
\]

\[
= f(x_0) + \sum_{n=1}^{\infty} \frac{h^n}{n!} f^{(n)}(x_0)
\]

where \( n! \) denotes the factorial of \( n \) and \( f^{(n)} \) denotes the \( n^{\text{th}} \) derivative of \( f \).

Truncation. If instead of evaluating the function for every possible \( n \), we stop at some particular upper limit \( N \), the truncated Taylor series is given by

\[
f(x_0 + h) = f(x_0) + \left( \sum_{n=1}^{N} \frac{h^n}{n!} f^{(n)}(x_0) \right) + \frac{h^{N+1}}{(N+1)!} f^{(N+1)}(\xi)
\]

where \( \xi \) is some particular value in the interval from \( x \) to \( x + h \). Since \( \xi \) depends upon the function \( f \) and the particular values of \( x_0 \) and \( h \), we often instead write

\[
f(x_0 + h) = f(x_0) + \left( \sum_{n=1}^{N} \frac{h^n}{n!} f^{(n)}(x_0) \right) + O(h^{N+1})
\]

to show that the error of approximating \( f \) by the first \( N \) terms of the Taylor series is \( O(h^{N+1}) \), that is, the error is “of order” \( h^{N+1} \).
Here is a plot of the first few truncated Taylor series approximations of \( \sin(x) \), expanded about \( x_0 = 0 \):

\[
\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} + \ldots
\]

Note that for each \( N \), the approximation is most accurate in the vicinity of \( x = 0 \). Also, as \( N \) increases, the approximation error decreases significantly at points further from the origin.

**Two-variable Taylor series.** The concept can be extended to a function \( f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R} \). The first few terms are:

\[
\begin{align*}
f(x_0 + r, y_0 + s) &= f(x_0, y_0) + rf_x(x_0, y_0) + sf_y(x_0, y_0) + \frac{r^2}{2} f_{xx}(x_0, y_0) + rs f_{xy}(x_0, y_0) + \frac{s^2}{2} f_{yy}(x_0, y_0) + O( |r| + |s| )^3
\end{align*}
\]

where \( f_x \) denotes \( \frac{d}{dx} f(x, y) \), \( f_y \) denotes \( \frac{d}{dy} f(x, y) \), and so on.

**Ordinary differential equations.** An *ordinary differential equation* (ODE) is one of the form

\[
\frac{d}{dx} y = f(x)
\]
Given some initial condition \( y_0 = y(x_0) \), we would like to find the solution \( y(x) \) to the differential equation. Although closed-form solutions to the function \( y(x) \) exist for some types of functions \( f \), there is no general solution for all functions \( f \). Hence, often, we end up approximating the solution to \( y(x) \) by numerical methods.

We begin by discretizing the domain of the function and evaluating \( y(x) \) at a number of regularly spaced points \( x_1, \ldots, x_m \), where \( x_{i+1} = x_i + h \). Denote the values of the approximated function \( y(x_i) \) as \( y_i \). In general, the approximate value \( y_i \) is not equal to the true value \( y(x_i) \), as we will see in the next and subsequent sections. Now, our task is to come up with an algorithm or method to estimate \( y_{i+1} \) given \( y_i \).

**Euler’s Method.** The simplest such method is based upon truncating the Taylor series expansion of \( y(x) \) after the first derivative term. We have

\[
y(x_{i+1}) = y(x_i + h) = y(x_i) + hy'(x_i) + O(h^2)
\]

\[
y(x_i) + hf(x_i, y_i) + O(h^2)
\]

Euler’s Method works by starting with \( x_0 \) and \( y_0 = y(x_0) \) and defining

\[
y_{i+1} \leftarrow y_i + hf(x_i, y_i)
\]

The \( O(h^2) \) term above indicates that the error resulting from approximating \( y(x_{i+1}) \) in this manner is of order \( h^2 \). This is called the local truncation error, and tells us how much error we add at each step of the algorithm in going from \( y_i \) to \( y_{i+1} \). We might also be interested in examining the full truncation error, which tells us how far off we will be from the true value of \( y_m \) after \( m \) steps of the algorithm. Some clever analysis (see the other handout for this lab) reveals that the full truncation error of Euler’s Method is \( O(h) \).

How could we get better local truncation error – say, \( O(h^3) \) or higher order? We could continue by adding more terms of the Taylor series. For instance,

\[
y(x_{i+1}) = y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(x_i) + O(h^3)
\]

We can evaluate \( y''(x_i) \) in terms of \( f \) by applying the chain rule:

\[
y''(x_i) = \frac{d}{dx} f(x_i, y(x_i)) = f_x(x_i, y(x_i)) + f_y(x_i, y(x_i)) f(x_i, y(x_i))
\]

The big drawback to this “improved” approach is the requirement that we be able to evaluate the partial derivatives of \( f \), which can become tedious and complicated.

**Runge-Kutta Methods.** It turns out that some principled manipulation of variables will allow us to achieve \( O(h^3) \) local truncation error without ever evaluating partial derivatives of \( f \). We will call any method that does this a second-order Runge-Kutta method. Begin by assigning

\[
y_{i+1} \leftarrow y_i + h(ak_1 + bk_2)
\]
where

\[ k_1 = f(x_i, y_i) \]
\[ k_2 = f(x_i + \alpha h, y_i + \beta h k_1) \]

The trick here is to solve for values of the constants \( a, b, \alpha, \beta \) such that all of the Taylor terms of order \( h^2 \) cancel out. By doing a Taylor expansion of \( y(x_{i+1}) \), we see that

\[
y(x_{i+1}) = y(x_i) + hf(x_i, y_i) + \frac{h^2}{2} \left( f_x(x_i, y(x_i)) + f_y(x_i, y(x_i))f(x_i, y(x_i)) \right) + O(h^3)
\]

Suppressing the function notation on \( f \), we have

\[
y(x_{i+1}) = y_i + hf + \frac{h^2}{2} (f_x + f_y f) + O(h^3) \tag{2}
\]

Now, perform a two-variable Taylor expansion of \( k_2 \):

\[
k_2 = f + \alpha hf_x + \beta hf_y f + O(h^2)
\]

By plugging the above expansion into equation (1) and grouping terms with similar exponents on \( h \), we have

\[
y_{i+1} = y_i + hak_1 + hbk_2
\]
\[
= y_i + haf + hb (f + \alpha hf_x + \beta hf_y f + O(h^2))
\]
\[
= y_i + h(af + bf) + h^2 b (\alpha f_x + \beta f_y f) + O(h^3)
\]

The last line above is equal to equation (2) if the following conditions are met:

\[
a + b = 1
\]
\[
\alpha b = \frac{1}{2}
\]
\[
\beta b = \frac{1}{2}
\]

This gives us three equations in four unknowns, and hence there are multiple valid choices for the variables. One popular method is to use \( a = b = \frac{1}{2} \), and set \( \alpha = \beta = 1 \). Known as Heun’s Method, the algorithm is given by

\[
y_{i+1} \leftarrow y_i + \frac{h}{2} \left( f(x_i, y_i) + f(x_i + h, y_i + hf(x_i, y_i)) \right) \tag{3}
\]

The Midpoint Method sets \( a = 0, b = 1, \) and \( \alpha = \beta = \frac{1}{2} \). The algorithm is

\[
y_{i+1} \leftarrow y_i + hf \left( x_i + \frac{h}{2}, y_i + \frac{h}{2} f(x_i, y_i) \right) \tag{4}
\]
Higher-order Runge-Kutta Methods. The same type of analysis can yield higher-order Runge-Kutta methods. The fourth-order methods define

\[ y_{i+1} \leftarrow y_i + h(ak_1 + bk_2 + ck_3 + dk_4) \]

with \( k_1, \ldots, k_4 \) defined similarly to the previous section. One of the most common fourth-order Runge-Kutta methods is defined as

\[
\begin{align*}
    k_1 &= f(x_i, y_i) \\
    k_2 &= f(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1) \\
    k_3 &= f(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2) \\
    k_4 &= f(x_i + h, y_i + hk_3)
\end{align*}
\]

It has local truncation error of \( O(h^5) \).

Simultaneous systems of ODEs. Note that we can use the exact same methods to solve simultaneous systems of ODEs defined on more than one variable. Given a vector of variables \( y = [y^{(1)}, \ldots, y^{(k)}]^T \), and a set of equations of the form

\[
\frac{d}{dx} y^{(i)} = f_i(x, y^{(1)}, \ldots, y^{(k)})
\]

we can rewrite the set of equations in vector form such that

\[
\frac{d}{dx} y = f(x, y)
\]

Then we can straightforwardly apply the methods above. For instance, the Midpoint Method:

\[
y_{i+1} \leftarrow y_i + h f \left( x_i + \frac{h}{2}, y_i + \frac{h}{2} f(x_i, y_i) \right)
\]

EXERCISE

Consider the damped mass-spring system pictured below:
Let's compare Euler's Method and fourth-order Runge-Kutta with the analytic solution for this system. First, determine the analytic model. From a free-body diagram, we know

\[ \dddot{z} + 2\zeta\omega\dot{z} + \omega^2 z = 0 \]

where \( \omega = \sqrt{\frac{K}{m}} \) and \( \zeta = \frac{B}{2\sqrt{mK}} \). If we hypothesize the solution to be of the form

\[ z(t) = Ae^{st} \]

the characteristic equation is

\[ s^2 + 2\zeta\omega s + \omega^2 = 0 \]

Solving for \( s \), we find

\[ s = -\zeta\omega \pm \omega\sqrt{\zeta^2 - 1} \]

Assume that \( \zeta > 1 \) so that the system is over-damped and \( s \) has two real roots. Then the solution for initial conditions \( z(0) = z_0 \) and \( \dot{z}(0) = \dot{z}_0 \) is given by

\[
\begin{align*}
  z(t) &= A_1 e^{s_1 t} + A_2 e^{s_2 t} \\
  s_1 &= -\zeta\omega + \omega\sqrt{\zeta^2 - 1} \\
  s_2 &= -\zeta\omega - \omega\sqrt{\zeta^2 - 1} \\
  A_1 &= \frac{s_2 z_0 - \dot{z}_0}{s_1 - s_2} \\
  A_2 &= \frac{s_1 z_0 - \dot{z}_0}{s_1 - s_2}
\end{align*}
\]

The dynamics in state variable form are given by

\[
\dot{q} = \begin{bmatrix} \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K}{m} & -\frac{B}{m} \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix} = Mq
\]

In terms of the methods above, set \( x = t \) and \( y = q \), so

\[ \frac{\partial}{\partial t}q = f(t, q) = Mq \]

Let's set \( K = 10 \text{ N/m}, m = 0.1 \text{ kg}, \) and \( B = 2.75 \text{ Ns/m} \), and start with the initial conditions \( z_0 = 1 \text{ m} \) and \( \dot{z}_0 = 0 \text{ m/s} \). Now compare Euler's method and fourth-order Runge-Kutta with step sizes of \( h = 0.05 \text{ s} \) and \( 0.10 \text{ s} \):
As is evident in the figure, the fourth order Runge-Kutta method with a step size of 0.05 is the most accurate, with no apparent divergence from the true function value. With a larger step size, Runge-Kutta initially has a large offset, but converges on the true position as the mass approaches the origin. Euler’s method with small step size has a steady offset from the analytic function, and with a larger step size, becomes unstable, eventually “blowing up” with ever-increasing oscillations.

**EXPERIMENT AND ANALYSIS**

Here is the coupled pendula system from the last lab:
The dynamics are given by
\[
\begin{align*}
ml^2 \ddot{\theta}_1 + mgl\theta_1 + k\ell^2 (\theta_1 - \theta_2) &= 0 \\
ml^2 \ddot{\theta}_2 + mgl\theta_2 + k\ell^2 (\theta_2 - \theta_1) &= 0
\end{align*}
\]

Create numerical approximations to the solution of the system dynamics, for each of the three cases we looked at last week:

1. Same initial angles, no initial velocity
2. Opposing initial angles, no initial velocity
3. One pendulum with non-zero initial angle, one with zero initial angle, zero initial velocity

You should model the system using both Euler’s Method and a fourth order Runge-Kutta method.

LABORATORY REPORT

1. Consider the case of a differential equation \( y' = f(x, y) \) where \( f \) does not depend upon \( y \); that is, the differential equation can be written as \( y' = f(x) \). In this case, numerically approximating the solution \( y(x) \) from \( x_0 \) to \( x_m \) is the same as approximating the definite integral of \( f \) from \( x_0 \) to \( x_m \). Describe, verbally and pictorially, what integration rules are implied by the second order Runge-Kutta methods defined by equations (3) and (4). It may be useful to consider the formulas for the area of a rectangle and the area of a trapezoid.

2. When deriving analytic models of dynamics, we often pay particular attention to questions about whether the system is overdamped, underdamped, or critically damped because the answers shape the analytic solution. Furthermore, we also make simplifying assumptions such as neglecting friction and \( \sin(\theta) \approx \theta \). Discuss why neither type of concern is as important when using numerical approximation methods as opposed to deriving analytic solutions.

3. In your numerical simulations of the coupled pendula, what is the maximum step size that can be used with Euler’s Method before it becomes unstable? What about fourth order Runge Kutta? Does the step size depend upon which case (1-3 from above) is being modeled?

4. Include tables and plots in your results section showing results from the analysis and numerical simulation for Cases 1-3.

5. Include a comparison of the results and errors from the analysis and simulation for Cases 1-3 in your discussion section.

6. Describe how you implemented the numerical simulation. You may include source code in an appendix.